

Claims

What is claimed is:

1. A system that test loads a server comprising:
a dynamic load adjustor component that dynamically adjusts user characteristics, for distribution thereof as a percentage of total requests sent to a server being load tested.
2. The system of claim 1 further comprising a profile characteristic data store that supplies the dynamic load adjustor component with weighting for a characteristic defined in a user profile.
3. The system of claim 2, the dynamic load adjustor component further comprises a weighting designator that randomly assigns to users characteristics based on weightings defined in the user profile.
4. The system of claim 2, the characteristic is at least one of: network connections, browser types, and load patterns.
5. The system of claim 2, the characteristic is statistically determined based on web log records.
6. The system of claim 2, the characteristic is predetermined in a single user profile.
7. The system of claim 1, further comprising a load coordinator component that adjusts an intensity of a load test based on a current distribution of users entering and leaving the server relative to a desired test load.
8. The system of claim 1, further comprising an artificial intelligence component.
9. The system of claim of claim 1 further comprising a closed loop control to enable a continual and sustained rate of requests to the server.

10. A system that stresses a server, comprising:
an execution engine that generates a scenario that loads the server *via* a plurality of users, the plurality of users is dynamically adjusted based on predetermined weightings of a user profile having weighted characteristics therein, wherein the scenario distributes user characteristics as a percentage of total requests.
11. The system of claim 10, the scenario comprises at least one of a test mix and a load profile.
12. The system of claim 10 further comprising a control input that adjusts rate of requests loaded onto the server.
13. The system of claim 10 further comprising a queuing mechanism that retrieves and sorts requests to be sent to the server.
14. The system of claim 10 further comprising a scheduler that determines number of requests to be generated for an upcoming period.
15. The system of claim 10 the requests are sorted according to a time function for execution.
16. A method for load testing a server comprising:
assigning weights to user characteristics in a user profile;
dynamically adjusting the user characteristics during the testing of the server; and
distributing the user characteristics as a percentage of total requests sent to the server.

17. The method of claim 16 further comprising comparing a current load on the server with a desired load.
18. The method of claim 17 further comprising creating a new user if the current load is less than a desired load.
19. The method of claim 17 further comprising reducing the current load by one upon ending an iteration, if the current load is not less than the desired load.
20. The method of claim 16 further comprising controlling a rate of loading *via* a feedback loop control.
21. A system for test loading a server comprising:
 - means for dynamically adjusting user characteristics while loading the server; and
 - means for distributing the user characteristics as a percentage of total requests sent to the server.